

AD-A185 781

ON THE WORK NEEDED TO FACTOR A SYMMETRIC POSITIVE
DEFINITE MATRIX(U) CALIFORNIA UNIV BERKELEY DEPT OF
INDUSTRIAL ENGINEERING AND O. M D CARVALHO JUN 87

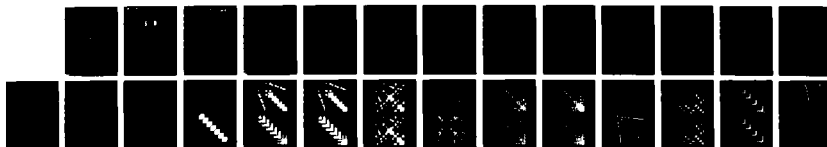
1/1

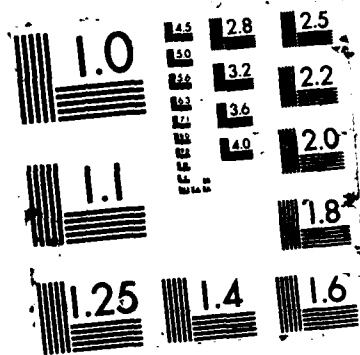
UNCLASSIFIED

ONR-87-14 N00014-87-K-0202

F/G 12/4

NL





AD-A185 781

DTIC
ELECTE
NOV 13 1987
S D

ON THE WORK NEEDED TO FACTOR A SYMMETRIC
POSITIVE DEFINITE MATRIX*

by

Marcio de Carvalho**

ORC 87-14

June 1987

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

*This research was supported in part by the Office of Naval Research under contract ONR N00014-87-K-0202, with the University of California, and by the Post-Graduate Education Federal Agency of Brazil, grant# 3817/82-4. Reproduction in whole or part is permitted for any purpose of the U. S. government.

**Industrial Engineering and Operations Research Dept., University of California Berkeley, Berkeley CA 94720

87 10 27 054

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ORC 87-14	2. GOVT ACCESSION NO. ADA185781	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) On the Work Needed to Factor a Symmetric Positive Definite Matrix		5. TYPE OF REPORT & PERIOD COVERED Technical
7. AUTHOR(s) Marcio de Carvalho		6. PERFORMING ORG. REPORT NUMBER ORC 87-15
9. PERFORMING ORGANIZATION NAME AND ADDRESS Operations Research Center 3115 Etcheverry Hall U.C. Berkeley Berkeley, CA 94720		8. CONTRACT OR GRANT NUMBER(s) N00014-87-K-0202
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Dept. of the Navy Arlingotn, VA 22217		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE June 1987
		13. NUMBER OF PAGES 24
		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Distribution Unlimited; Approved for public release		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) (see report)		

DD FORM 1473
1 JAN 75EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102- LF-014-6601UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A

On the Work Needed to Factor a Symmetric Positive Definite Matrix

Marcio de Carvalho †

Department of Industrial Engineering and Operations Research,
University of California, Berkeley, CA 94720

ABSTRACT

When comparing different row ordering strategies, the measure often used is the *fill-in*, or the number of additional non-zeroes elements. In this report, we propose another measure: the number of arithmetic operations necessary to factor the matrix. Two classical ordering strategies: *Minimum Degree* and *Minimum Local Fill-in* are compared with respect to this measure and *Minimum Local Fill-in* usually produces better results than *Minimum Degree*. Also, an application is presented where the number of arithmetic operations may be more interesting measure of performance is presented: Karmarkar's Linear Programming Algorithm.

Keywords: heuristic methods;

Key Words: Numerical Linear Algebra; Matrix Computations; Graph Theory, Linear Programming.

August 26, 1987



Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

On the Work Needed to Factor a Symmetric Positive Definite Matrix

Marcio de Carvalho †

Department of Industrial Engineering and Operations Research,
University of California, Berkeley, CA 94720

1. Introduction

We are interested in studying the influence of the ordering of the rows and columns of a symmetric positive definite matrix A on the work (number of arithmetic operations) required to obtain its Cholesky factors. Put in other words, we want to compare the effect of the use of different permutation matrices P on the work necessary to obtain a lower triangular matrix L such that $PAP^T = LL^T$.

The complexity of the problem of obtaining an ordering which minimizes the work is not known; it is conjectured by this author that this problem is at least as hard as the NP-hard problems. A strong indication of this difficulty is that a simpler but somewhat related problem is known to be in this class, namely the problem of computing an ordering which minimizes the fill-in [Yannakakis 1981]. A practical implication of the fact that a problem is NP-hard is that no efficient algorithm for its solution is known. And if one devises an algorithm for any problem in this class all others would also be solvable in an efficient way by this same algorithm, which makes the existence of such an algorithm very unlikely.

In our specific case, being NP-hard means that the effort of finding an ordering which minimizes any of the above mentioned criteria is greater than that required for the solution of the system itself, and therefore it would be a waste effort to obtain it exactly.

Heuristics have been devised to obtain an ordering which will fulfill approximately a desired criterion and would not be computationally expensive. Using an heuristic to solve a problem presents some drawbacks since the computed solution is not necessarily a good one. There have been devised a number of ways to measure the performance of a strategy and an accepted measure for ordering heuristics is to submit it to a series of standard test problems and from the data gathered, derive conclusions. This will be the approach used here.

† on leave from DCC-ICEx, Universidade Federal de Minas Gerais, Belo Horizonte, Brasil.

A widely used ordering heuristic is known as *Minimum Degree*, a description can be found in [Rose 1973]. Experience has shown that it presents a good trade-off between cost of computation and reduction of fill-in [Duff et al 1986]. Another heuristic is *Minimum Local Fill-in*, or minimum deficiency also presented in [Rose 1973]. The computation of this ordering is more expensive than *Minimum Degree*, but because it uses more information during the computation it usually gives a smaller fill-in. *Minimum Local Fill-in* does not present empirically as good as a trade-off between execution time and reduction of fill-in as *Minimum Degree*, for this reason it has not been as widely used as the former heuristic. It is pertinent to notice that since these are only heuristics, examples can be constructed where *Minimum Degree* yields a smaller fill-in than *Minimum Local Fill-in* and vice-versa [Duff et al 1986].

The main objective of this report is to compare the performance of these two classical ordering heuristics using as a measure, the work necessary to factor the ordered matrix. A natural question is whether the minimization of the fill-in also minimizes the work per solution; this will be addressed in the next section.

We implemented a version of *Minimum Local Fill-in* based on the description in [Vlach and Singal 1983] and it is compared to a *Minimum Degree* implementation from the YALE package on some standard test problems, obtained through electronic mail. The results are presented in section 3. And finally in section 4, the conclusions are presented. In the appendix, the tables and some pictures of matrices are presented.

2. Some Theoretical Facts

In this section we will present some facts relating vertex ordering to fill-in and factoring work. First, some definitions and notation.

Definition 1

$G^A = \{X^A, E^A\}$ is the graph associated to the symmetric matrix A . The vertices of X^A will correspond to rows/columns of A , and there will be an edge connecting vertex i to vertex j , $e_{ij} = e_{ji} \in E^A$ if there is a non-zero element in row i and column j . Let's denote by n , the cardinality of the set X^A , which is the number of rows/columns of A .

Let α be an ordering of the vertices of G^A , such that $\alpha(i)$ is the i^{th} vertex to be eliminated and let

G^α be the graph obtained after the elimination of the vertices of G^A according to the sequence α . The graph G^α will have the same vertex set as G^A and its edge set will be the union of the original edges E^A and the edges created by the elimination process $E^{F(\alpha)}$, called the fill-in edges.

Definition 2

$d_{\alpha(i)}$ is the degree of the i^{th} vertex in the ordering α .

Definition 3

$work(\alpha) = \sum_{i=1}^n d_{\alpha(i)}^2$ This is asymptotic number of arithmetic operations necessary to perform the factorization of the matrix A . An interested reader will find the exact expression in [Rose 1973].

Fact 1

Given two orderings α and β , such that $E^\alpha = E^\beta$ then $w(\alpha) = w(\beta)$.

Note that $G^\alpha = G^\beta$ is a chordal graph and α and β are two perfect elimination orderings since the factoring of G^α following the ordering α does not introduce any fill-in. By [Rose 1983], we have that:

$$\left\{ d_{\alpha(1)}, d_{\alpha(2)}, \dots, d_{\alpha(n)} \right\} = \left\{ d_{\beta(1)}, d_{\beta(2)}, \dots, d_{\beta(n)} \right\}$$

Or in words, for a chordal graph, the set of degrees encountered during the factorization following any perfect elimination ordering is the same. And therefore, the work is the same.

Maybe a more interesting case is when the number of added edges for two different orderings is the same and some of these edges connects different vertices. This is examined next.

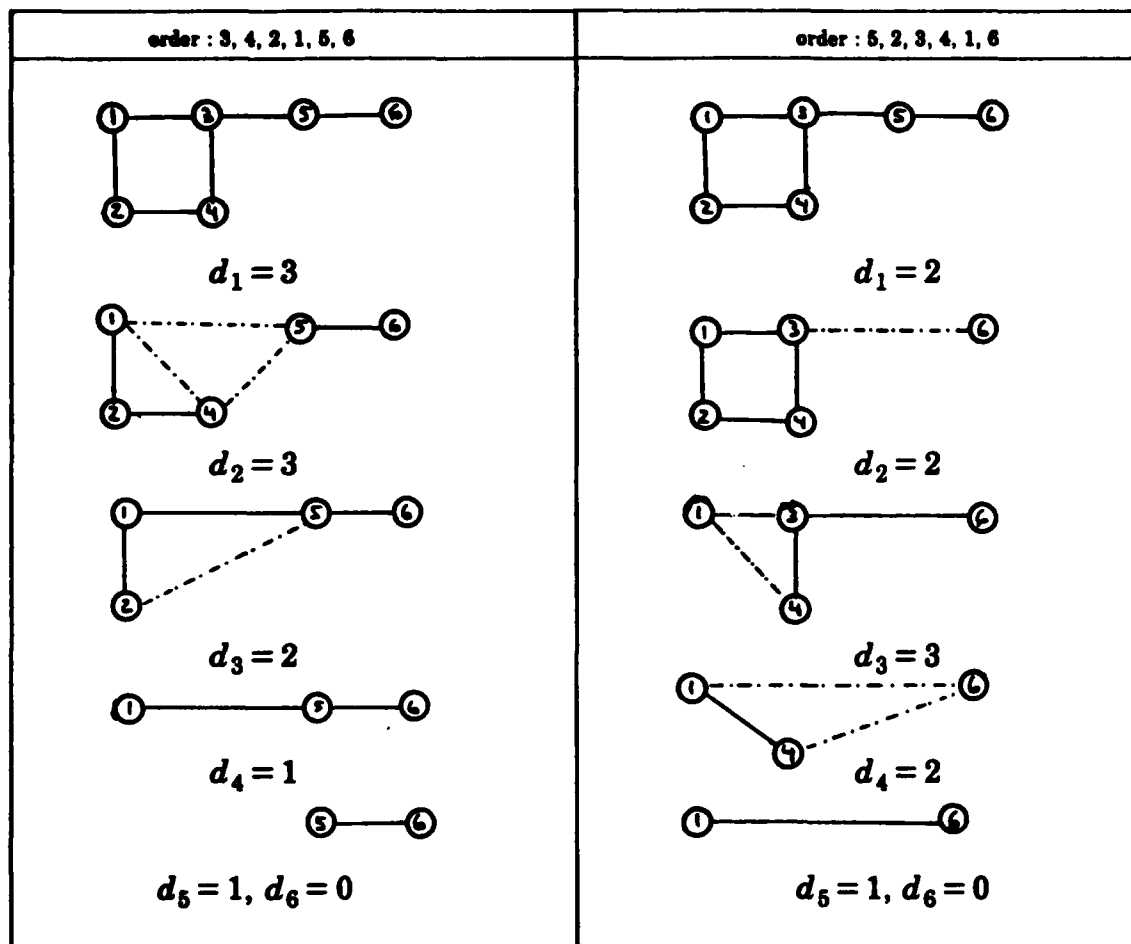
Fact 2

Given two orderings α and β , such that $|E^\alpha| = |E^\beta|$ then it is possible that $w(\alpha) \neq w(\beta)$.

This is shown by the example I, following:

From Fact 2, one sees that fill-in alone might not be the best measure of performance of an heuristic. It is possible to have the same number of fill-in edges and different work. From these facts, it not clear that *Minimum Degree* is better than *Minimum Local Fill-in* with respect to work per factoring.

A further question that remained unanswered concerns the case where the orderings are minimizers of the fill-in. Are the above results still true? How different can the graphs G^γ and G^δ be, when γ and δ are orderings that minimize the fill-in? We have been able to construct examples where the resulting graphs are very different, but in all of them, the degree sequence has been the same. Is this true in general? If one is just interested in the practical aspects in the application of the orderings, one should not be



Example 1

disturbed by these questions. An optimal ordering is too hard to be computed.

3. Empirical Results

A first difficulty in comparing *Minimum Degree* and *Minimum Local Fill-in* was the lack of code for *Minimum Local Fill-in* due to its reduced acceptance by the community. The solution was to implement our own version. A brief bibliographic research revealed just [Vlach and Singal 1983] as a source of *Minimum Local Fill-in* implementation "tricks"; they also present in the appendix a FORTRAN implementation of the heuristic. Unfortunately, after a superficial look at their code two mistakes were discovered, so rather than fixing them, we decided on implementing our own, but still using their ideas. As a measure of complexity of the algorithm, one may use the number of non-comment lines of the source code. The final FORTRAN version of *Minimum Local Fill-in* had about 300 lines of non-comments, which is comparable to the 220 of HARWELL's MA17E and 230 of YALE's MD.

The performance of the two available *Minimum Degree* codes were comparable. The MD routine of the YALE package was marginally better, and it was the one used as a reference.

The matrices used as test case are constructed from Linear Programming (LP) problems obtained through electronic mail from *netlib@anl-mcs.arpa*, Argonne National Laboratories, Illinois. If we denote the LP matrix by B , the test matrices used in this report are BB^T . Incidentally, this is the format used by Karmarkar's LP solver.

Table I presents Cholesky Factors Statistics for the matrices. The first column is the LP problem name as given by *netlib*. In this table and all the following ones, the LP problems are presented in nondecreasing order with respect to the number of nonzeros in the original LP matrix B . In the second column, Rows, contains the size of the square symmetric positive definite matrix BB^T . The third column, NZ, contains the initial number of Non-Zeros of half of the matrix BB^T . The next three columns are relative to *Minimum Degree* and the last three concerns *Minimum Local Fill-in*. Fill contains the fill-in in each Cholesky factor, Ops is the number of arithmetic operations necessary to compute the factor and finally time is the time in seconds of IBM 3090 necessary to compute the ordering.

Table II summarizes Table I, presenting the relative percentual change of the values from Table I. For each quantity, the value relative to *Minimum Local Fill-in* was subtracted from the correspondent *Minimum Degree* value and divided by the value for *Minimum Degree*. This result was then multiplied by one hundred.

The data presented does not contradict the established dominance of *Minimum Degree* over *Minimum Local Fill-in*, showing that, in average, the percentual reduction in fill-in (and work) achieved by *Minimum Local Fill-in* does not balance with the increase on processing time with respect to *Minimum Degree*. By examining the columns of fill-in and operations, we note that there is no clear relation between these two quantities, for some cases, the reduction of work was greater than the reduction of fill-in and in some cases, the opposite is observed.

These results seem to imply that one should use *Minimum Degree* when decomposing the matrix only once and that *Minimum Local Fill-in* should be the choice when a sequence of matrices with the same non-zero structure are to be factored. An example of application where *Minimum Local Fill-in* might be the choice is found in Karmarkar's algorithm for LP, where at each iteration k , a matrix of the form BD_kB^T needs to be factored. The matrix D_k is a diagonal matrix that is a function of the iteration k , see [Adler et al 1987]. Note that only the values of the matrix to be factored changes at each iteration, and the non-zero structure is the same during the whole process.

The same problems of Table I were solved by Karmarkar's algorithm, using both ordering techniques and the results are presented in Table III. The number below *Min Fill* and *Min Degree* are the solution times on the IBM 3090. The value in the column *% Change* was computed using the same algorithm previously described. Note that *Minimum Local Fill-in* gets better as the problem size increases.

One might argue that the YALE code is not as up-to-date as our *Minimum Local Fill-in* implementation. To address this point, Table IV was constructed. Here the ordering time was subtracted from the total solution time. Still one can see the decrease on solution time of the matrix ordered by *Minimum Local Fill-in* as the problem size increases.

In the appendix, pictures of some matrices here treated are presented, they provide a nice way to visualize the effect of the different orderings.

4. Conclusions

The number of arithmetic operations necessary to factor a symmetric positive definite matrix or work, as it is referred here in this report, is an important measure of quality of an heuristic and has not been much considered in the literature. Its importance can be noted specially when a sequence of factorizations is to be performed on matrices with the same non-zero structure. In this case, any reduction on work will be multiplied by the number of factorizations.

In this report, we use the measure of work to compare two well known ordering heuristics and we observe that matrices obtained after *Minimum Local Fill-in* will, in most of the cases, be factored in less time than the ones after *Minimum Degree*. But because the larger processing time for *Minimum Local Fill-in*, the total factoring time will only be smaller if a number of factorizations are to be performed. This is the case observed in the implementation of Karmarkar's algorithm here presented.

5. Acknowledgments

The work presented in this report was inspired by the lectures of Prof. Beresford Parlett during the Spring '87 edition of Sparse Matrices Methods. The author would like to thank his encouragement to write this up. I would like to thank Prof. Doug Shier for interesting and inspiring discussions on the theory of Chordal Graphs which brought up many of the questions presented here. The comparison of the ordering strategies in the Linear Programming Algorithm could not be done without the help and ideas of Ilan Adler, Mauricio G.C. Resende and Geraldo Veiga.

6. References

- Adler I., N. Karmarkar, M. Resende and G. Veiga, "An Implementation of Karmarkar's Algorithm for Linear Programming", to appear in *Math. Programming*, 1987.
- Duff, I., A. Erisman and J. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, 1986.
- Rose D., "A Graph-Theoretic Study of the Numerical Solution of Sparse Positive Definite Systems of Linear Equations", in *Graph Theory and Computing*, R. Read (ed.), Academic Press, 1973.
- Vlach J. and K. Singal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold Company, 1983.
- Yannakakis, "Computing the Minimum Fill-in is NP-Complete", *SIAM J. Alg. Disc. Meth.*, 2, 77-79, 1981.

7. Appendix**Tables and Pictures**

Cholesky Factors Statistics

Problem	Rows	NZ	Minimum Degree			Minimum Local Fillin		
			Fil	Ops	Time	Fil	Ops	Time
AFIRO	27	63	17	77	.00	17	77	.00
ADLITTL	55	322	27	1155	.01	27	1155	.01
SCAGR7	128	478	128	1410	.02	119	1343	.01
SHARE2B	96	775	155	4428	.00	106	4052	.02
SHARE1B	112	855	458	8809	.02	203	5035	.03
SCORPIO	360	1555	409	6458	.03	303	5609	.05
SCAGR25	470	1900	578	6414	.05	515	5969	.05
SCTAP1	300	1386	981	11976	.04	868	10251	.06
BRANDY	134	2056	660	35857	.07	547	31543	.12
SCSD1	77	1056	259	12119	.03	259	12119	.04
ISRAEL	174	11053	261	494040	.23	163	484575	.91
BANDM	246	2683	1185	40045	.07	740	28189	.12
SCFXM1	315	2828	1820	47211	.06	1090	31279	.13
E226	208	2475	731	32949	.07	725	32836	.13
SCRS8	456	1497	3181	41109	.08	3159	41024	.17
BEACONF	115	1605	7	16427	.11	8	16450	.05
SCSD6	147	1952	446	19611	.06	446	19611	.09
SHIP04S	249	2578	307	17985	.27	164	15833	.06
SCFXM2	630	5676	3485	89278	.13	2246	63508	.27
SHIP04L	325	3822	237	25068	.91	160	24077	.09
SHIP08S	334	3218	560	23699	.18	316	19815	.08
SCTAP2	1090	5505	8275	256907	.27	6555	156195	.57
SCFXM3	945	8524	5150	131345	.19	3402	95737	.40
SHIP12S	422	3811	830	29576	.07	451	23840	.09
SCSD8	397	3883	1599	35240	.04	1599	35240	.17
SCTAP3	1480	7386	10603	311467	.36	8845	203094	.81
CZPROB	689	5980	390	35432	6.06	393	35490	.23
25FV47	793	10922	22576	1201632	.62	16764	740991	2.58
SHIP08L	528	6244	356	40391	1.58	310	39798	.15
SHIP12L	692	8267	542	55294	1.51	476	54461	.18

Table I

% Changes				
Problem	Rows	Fillin	Ops	Time
AFIRO	27	0.0	0.0	0.0
ADLITTL	55	0.0	0.0	0.0
SCAGR7	128	7.0	4.8	50.0
SHARE2B	96	31.6	8.5	
SHARE1B	112	55.7	42.8	-50.0
SCORPIO	360	25.9	13.1	-66.7
SCAGR25	470	10.9	6.9	0.0
SCTAP1	300	11.5	14.4	-50.0
BRANDY	134	17.1	12.0	-71.4
SCSD1	77	0.0	0.0	-33.3
ISRAEL	174	37.5	1.9	-295.7
BANDM	246	37.6	29.6	-71.4
SCFXM1	315	40.1	33.7	-116.7
E226	208	0.8	0.3	-85.7
SCRS8	456	0.7	0.2	-112.5
BEACONF	115	-14.3	-0.1	54.5
SCSD6	147	0.0	0.0	-50.0
SHIP04S	249	46.6	42.0	77.8
SCFXM2	630	35.6	28.9	-107.7
SHIP04L	325	32.5	4.0	90.1
SHIP08S	334	43.6	16.4	55.6
SCTAP2	1090	20.8	39.2	-111.1
SCFXM3	945	33.9	27.1	-110.5
SHIP12S	422	45.7	19.4	-28.6
SCSD8	397	0.0	0.0	-325.0
SCTAP3	1480	16.6	34.8	-125.0
CZPROB	689	-0.8	-0.2	96.2
25FV47	793	25.7	38.3	-316.1
SHIP08L	528	12.9	1.5	90.5
SHIP12L	692	12.2	1.5	88.1

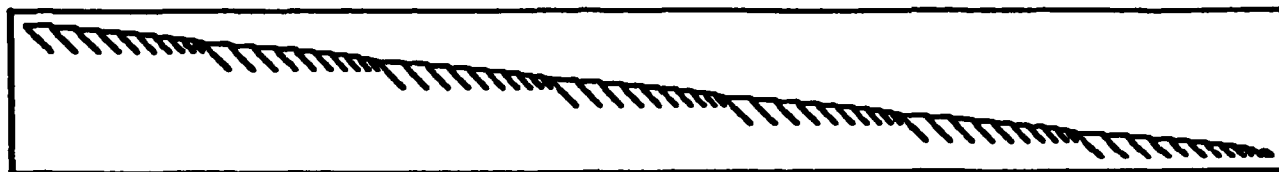
Table II

Solution Times					
Problem	Rows	Iterations	Min Fill	Min Degree	% Change
AFIRO	27	20	.05	.04	-25.0
ADLITTL	55	24	.13	.12	-8.3
SCAGR7	128	24	.18	.17	-5.9
SHARE2B	96	29	.32	.29	-10.3
SHARE1B	112	38	.47	.58	19.0
SCORPIO	360	24	.49	.51	3.9
SCAGR25	470	29	.70	.69	-1.4
SCTAP1	300	33	.80	.85	5.9
BRANDY	134	36	1.49	1.52	2.0
SCSD1	77	19	.44	.41	-7.3
ISRAEL	174	37	13.08	13.46	2.8
BANDM	246	30	1.26	1.54	18.2
SCFXM1	315	33	1.66	1.97	15.7
E226	208	34	1.58	1.56	-1.3
SCRS8	456	39	2.28	2.30	0.9
BEACONF	115	23	.62	.69	10.1
SCSD6	147	22	.84	.83	-1.2
SHIP04S	249	30	1.01	1.26	19.8
SCFXM2	630	39	3.83	4.38	12.6
SHIP04L	325	28	1.39	2.31	39.8
SHIP08S	334	32	1.35	1.58	14.6
SCTAP2	1090	34	5.52	6.71	17.7
SCFXM3	945	40	5.87	6.66	11.9
SHIP12S	422	35	1.75	1.91	8.4
SCSD8	397	23	1.82	1.68	-8.3
SCTAP3	1480	36	7.78	9.44	17.6
CZPROB	689	52	3.64	9.76	62.7
25FV47	793	54	31.70	46.17	31.3
SHIP08L	528	31	2.44	4.00	39.0
SHIP12L	692	32	3.43	4.89	29.9

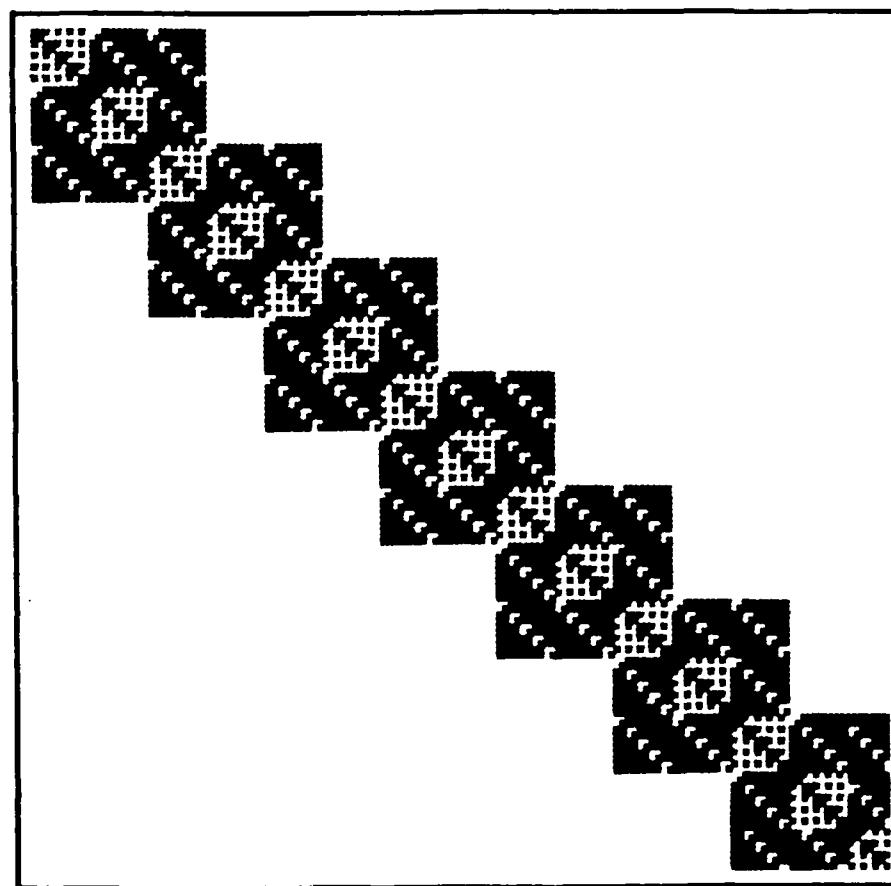
Table III

Solution Times Without Ordering Time			
Problem	Min Fill	Min Degree	% Change
AFIRO	0.05	0.04	-25.0
ADLITTL	0.12	0.11	-9.1
SCAGR7	0.17	0.15	-13.3
SHARE2B	0.30	0.29	-3.4
SHARE1B	0.44	0.56	21.4
SCORPIO	0.44	0.48	8.3
SCAGR25	0.65	0.64	-1.6
SCTAP1	0.74	0.81	8.6
BRANDY	1.37	1.45	5.5
SCSD1	0.40	0.38	-5.3
ISRAEL	12.17	13.23	8.0
BANDM	1.14	1.47	22.4
SCFXM1	1.53	1.91	19.9
E226	1.45	1.49	2.7
SCRS8	2.11	2.22	5.0
BEACONF	0.57	0.58	1.7
SCSD6	0.75	0.77	2.6
SHIP04S	0.95	0.99	4.0
SCFXM2	3.56	4.25	16.2
SHIP04L	1.30	1.40	7.1
SHIP08S	1.27	1.40	9.3
SCTAP2	4.95	6.44	23.1
SCFXM3	5.47	6.47	15.5
SHIP12S	1.66	1.84	9.8
SCSD8	1.65	1.64	-0.6
SCTAP3	6.97	9.08	23.2
CZPROB	3.41	3.70	7.8
25FV47	29.12	45.55	36.1
SHIP08L	2.29	2.42	5.4
SHIP12L	3.25	3.38	3.8

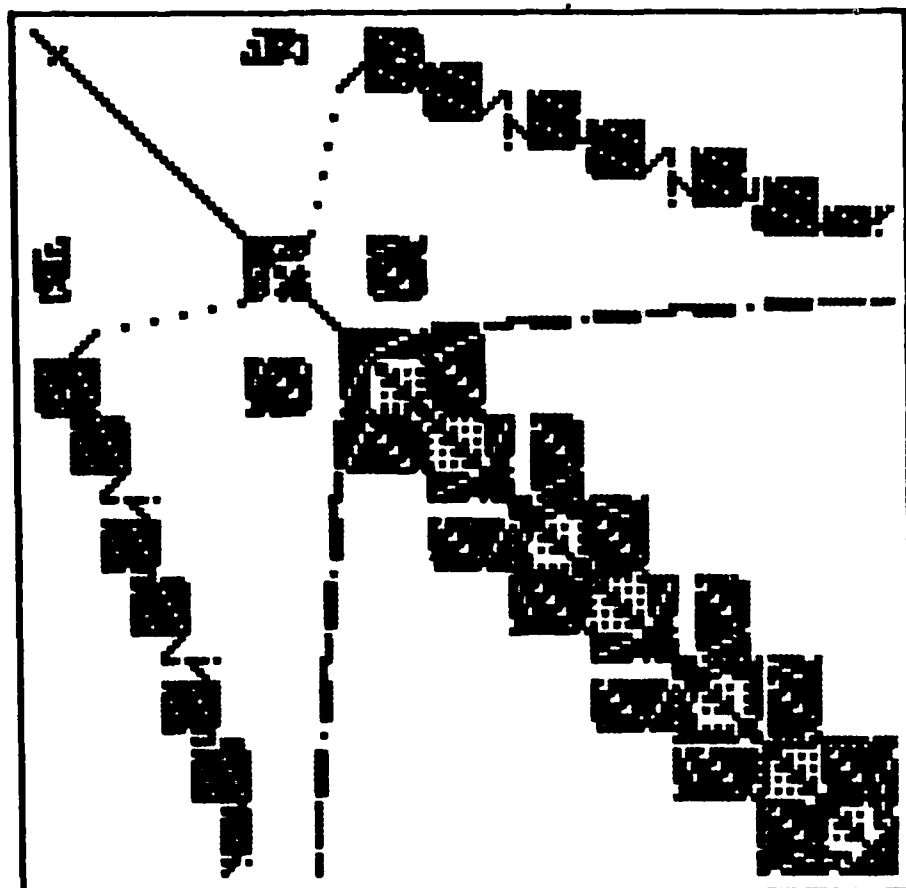
Table IV



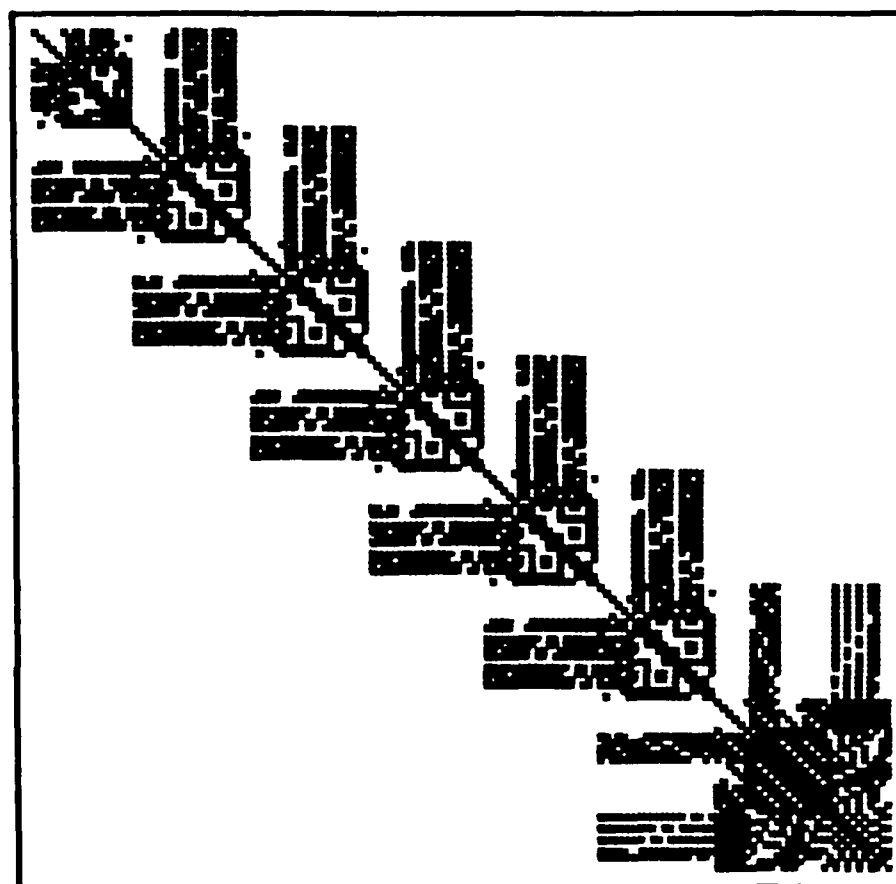
SCSD6 - LP matrix before cleaning



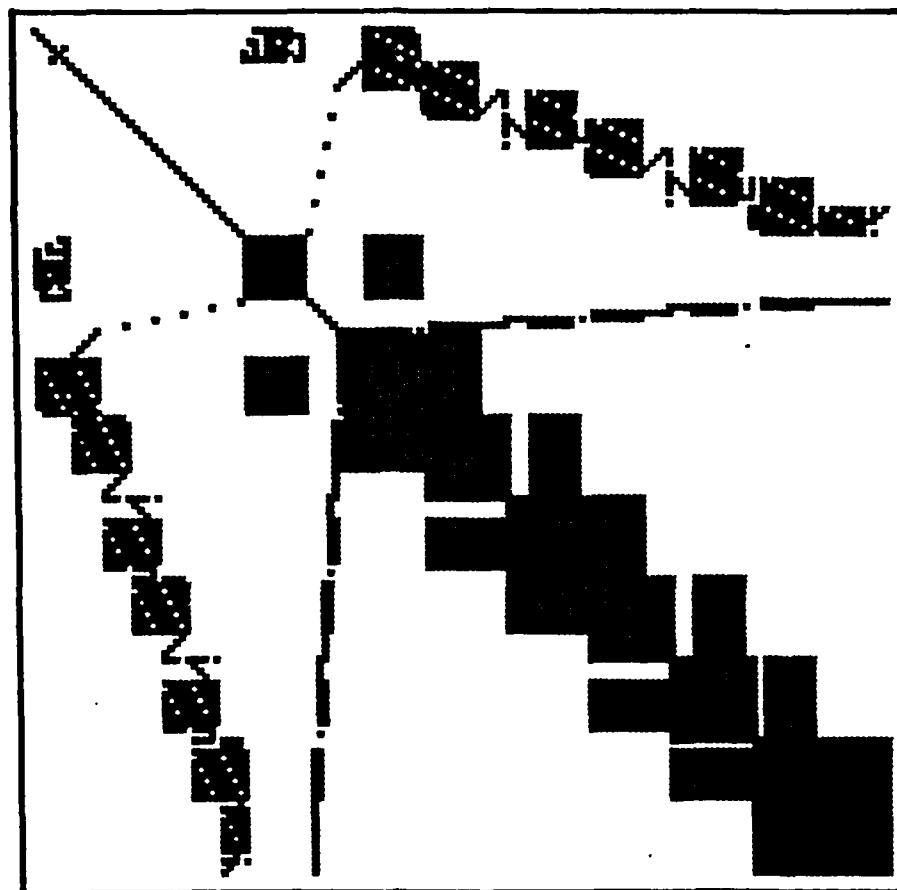
SCSD6 - AA^t before ordering



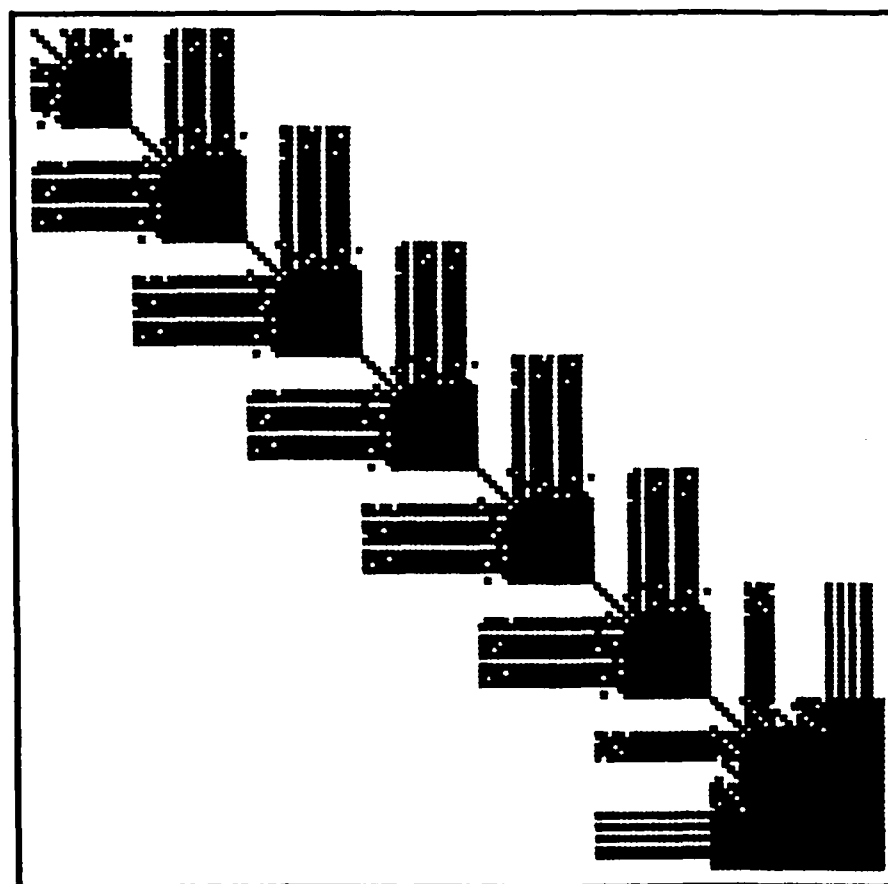
SCSD6 - AA^t after YALE ordering



SCSD6 - AA^t after Minimum Local Fill-in ordering



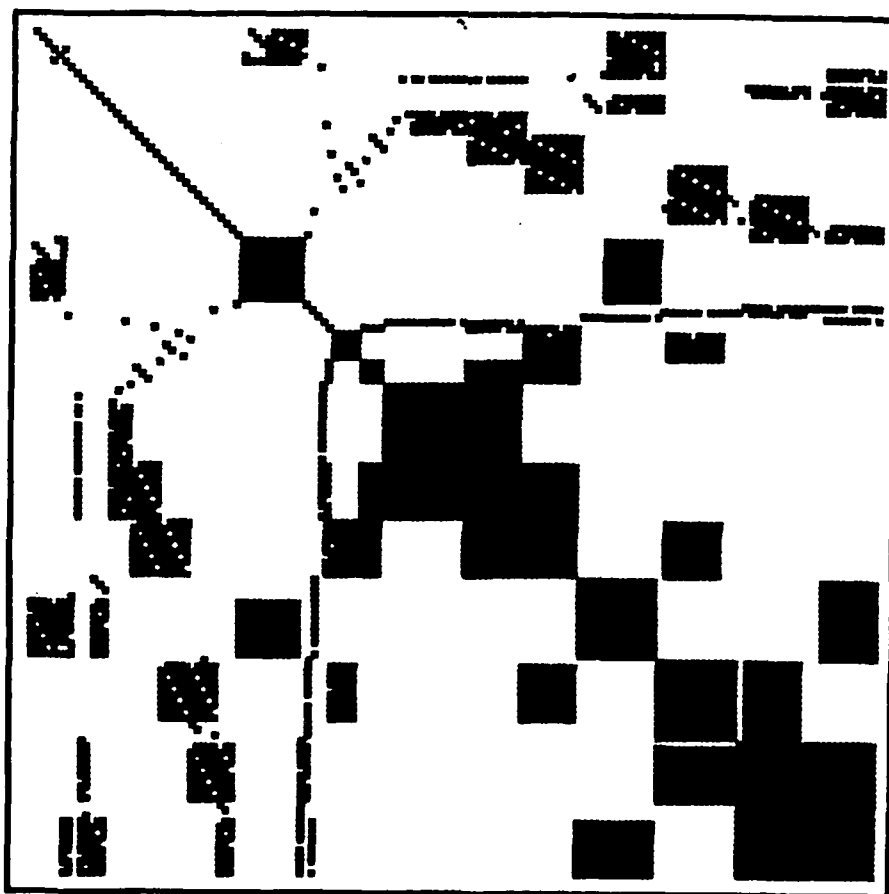
SCS06 - Cholesky factors after YALE ordering



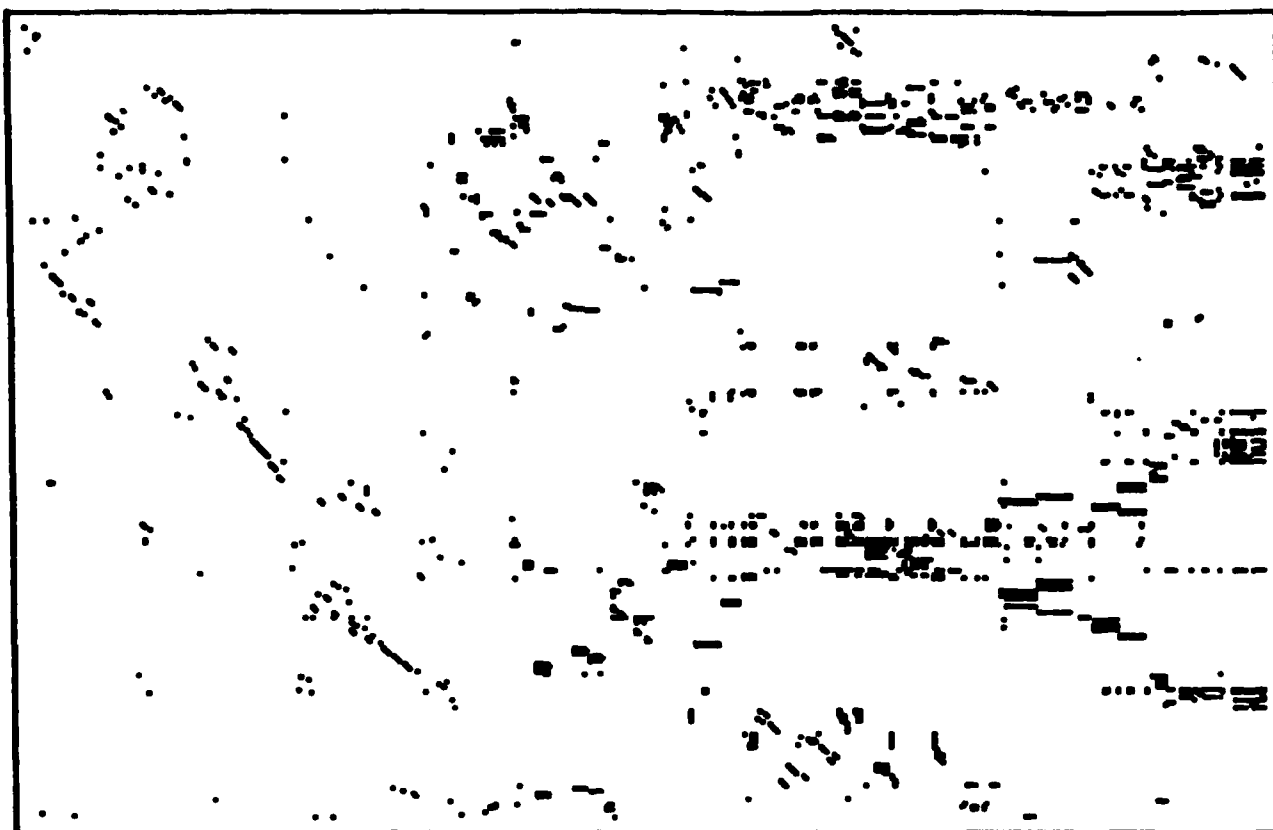
SCS06 - Cholesky factors after Minimum Local Fill-in ordering



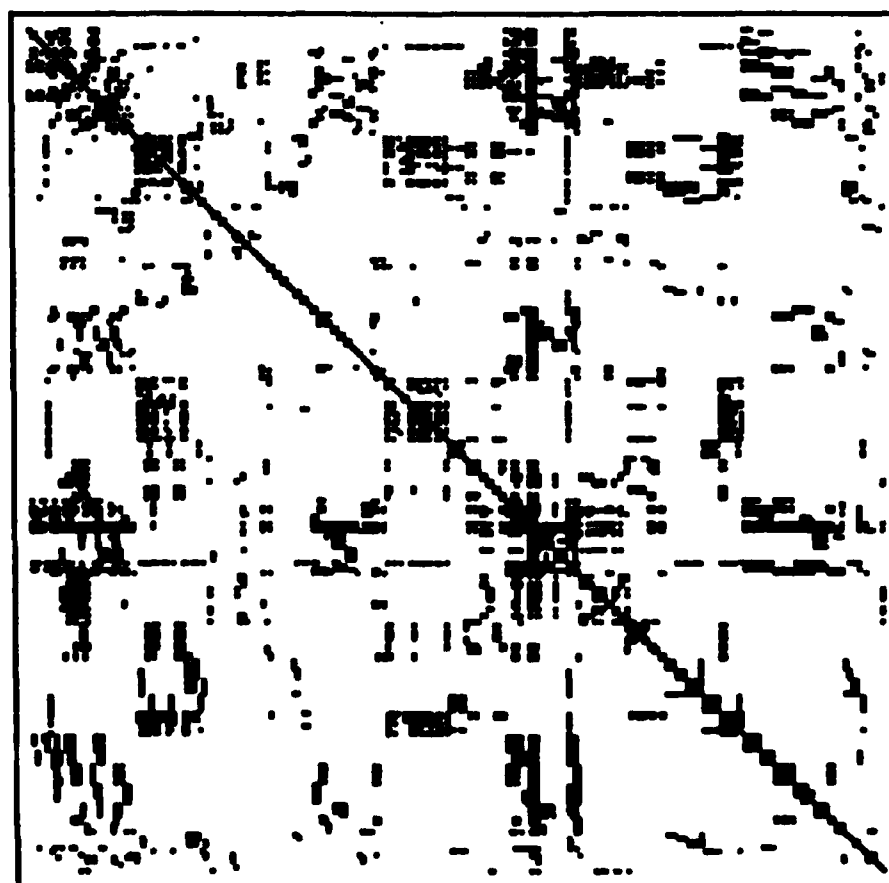
SCSD6 - AA^t after Harwell's minimum degree ordering



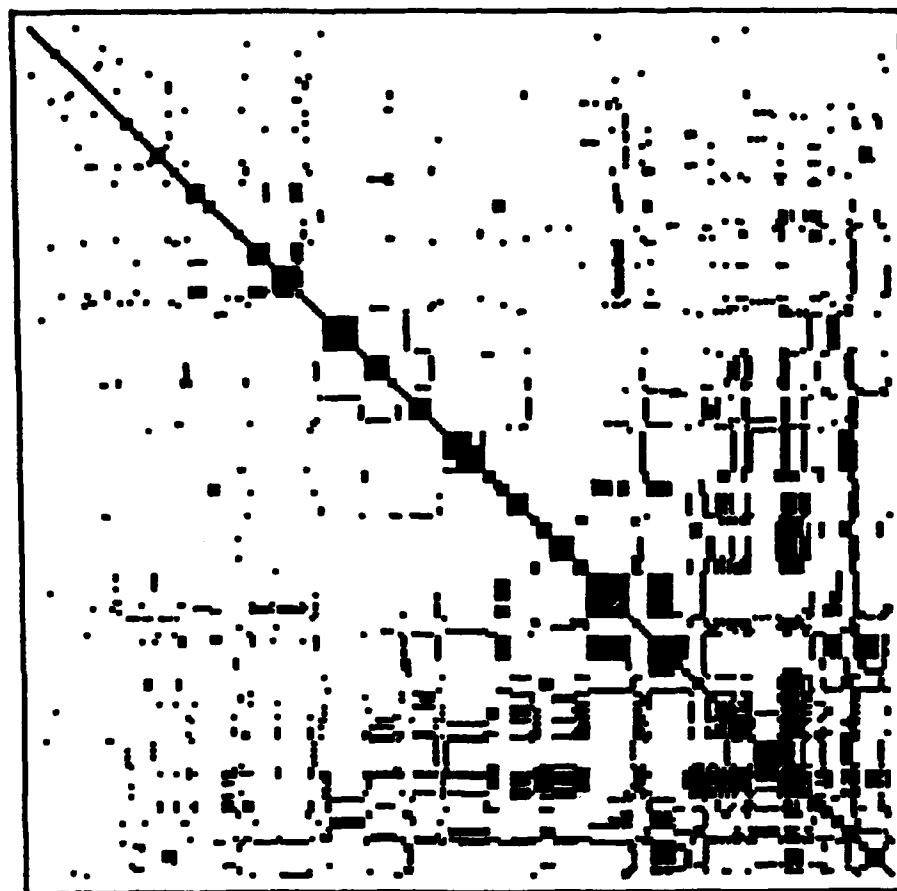
SCSD6 - Cholesky factors after Harwell's minimum degree ordering



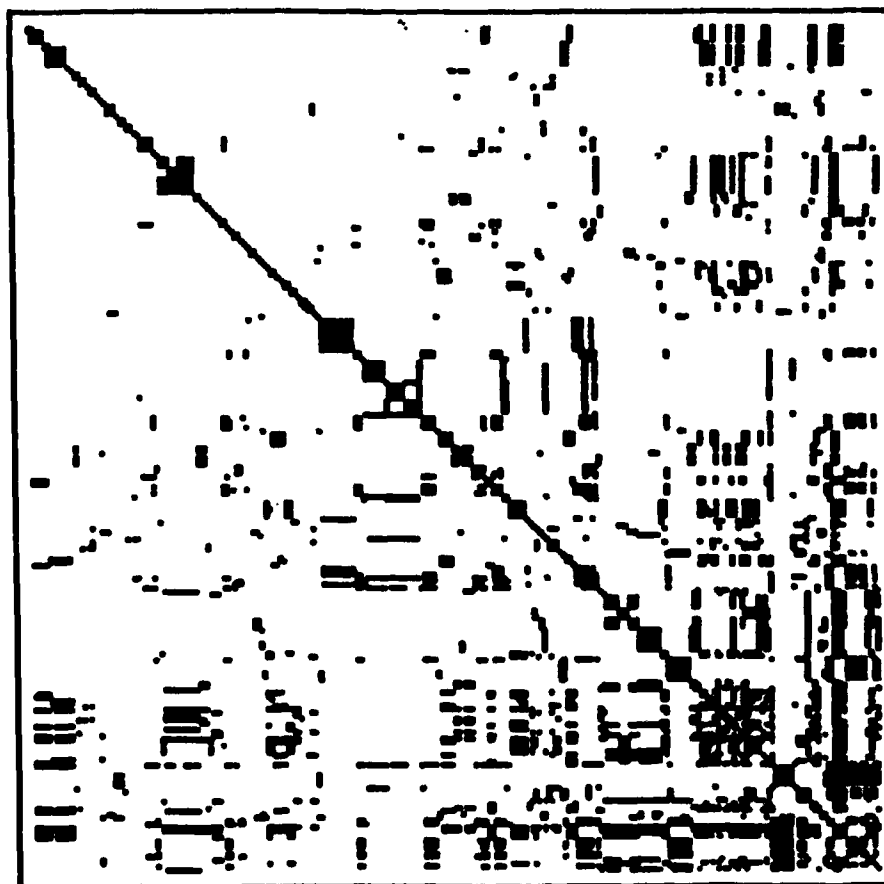
BANDM - LP matrix before cleaning



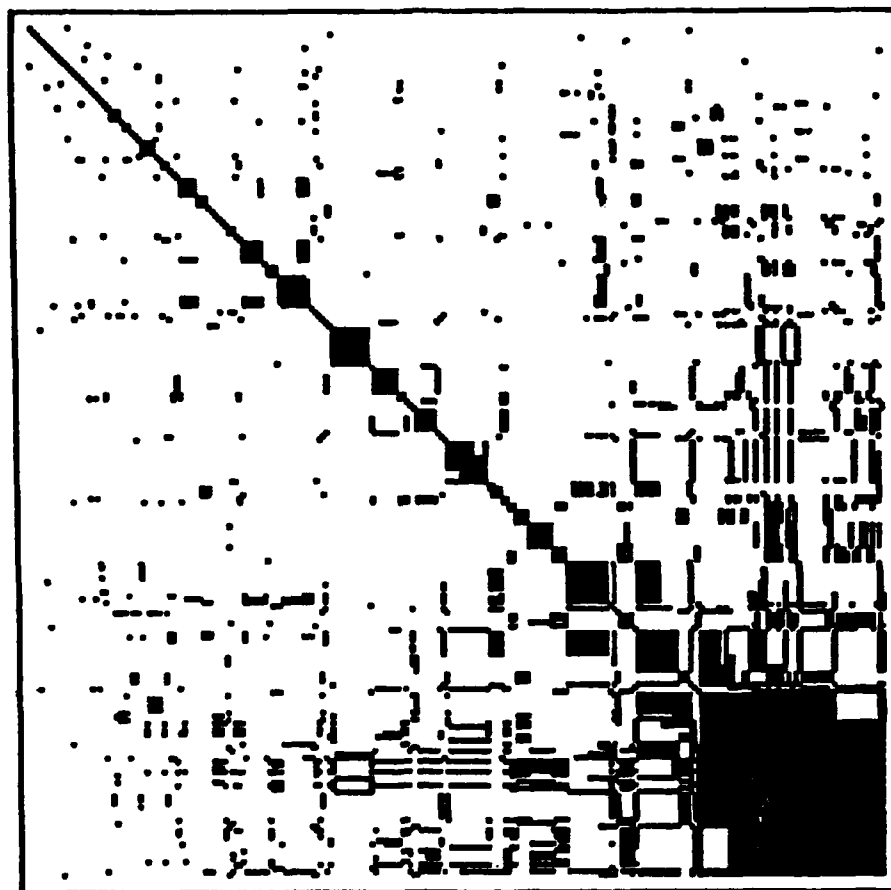
BANDM - AA^T before ordering



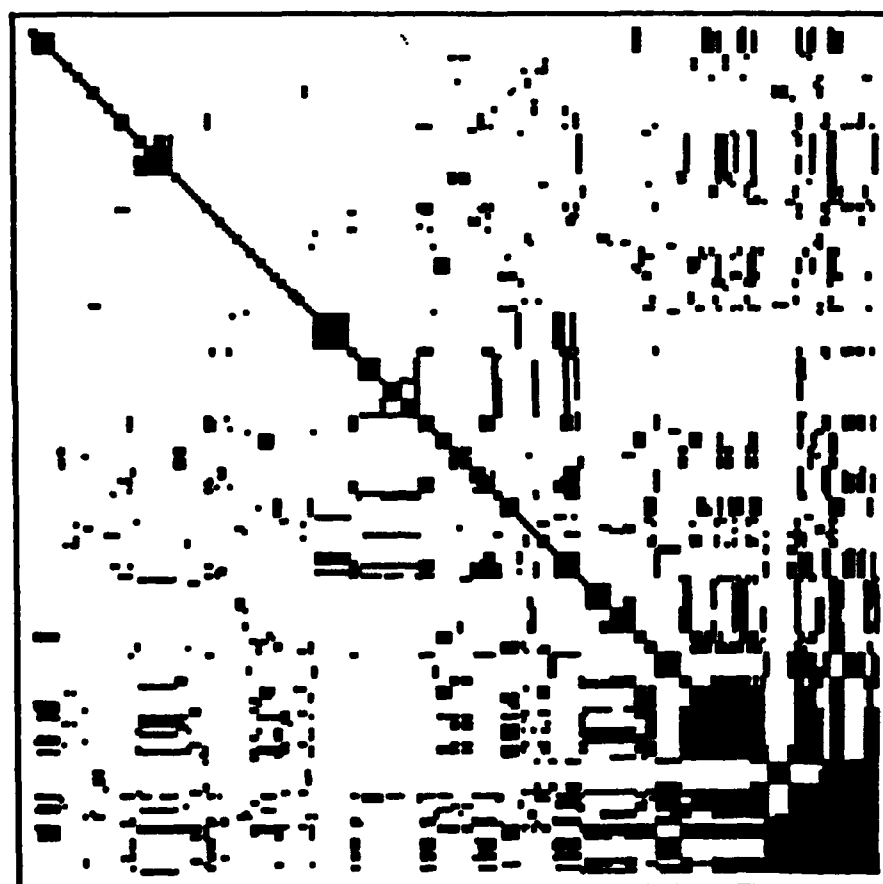
BANDM - AA^t after YALE ordering



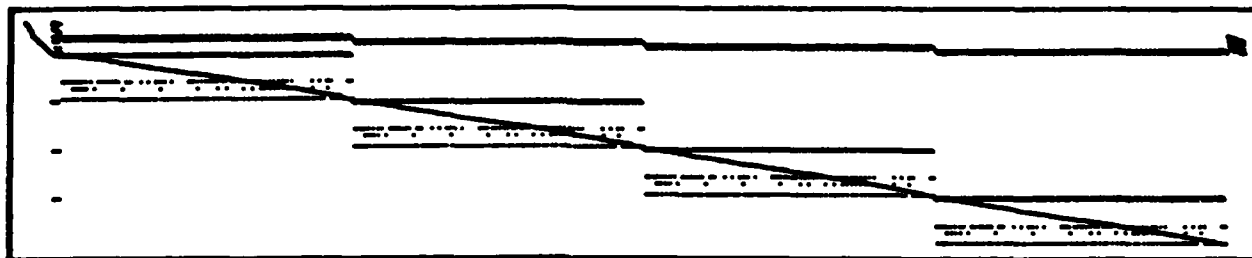
BANDM - AA^t after Minimum Local Fill-in ordering



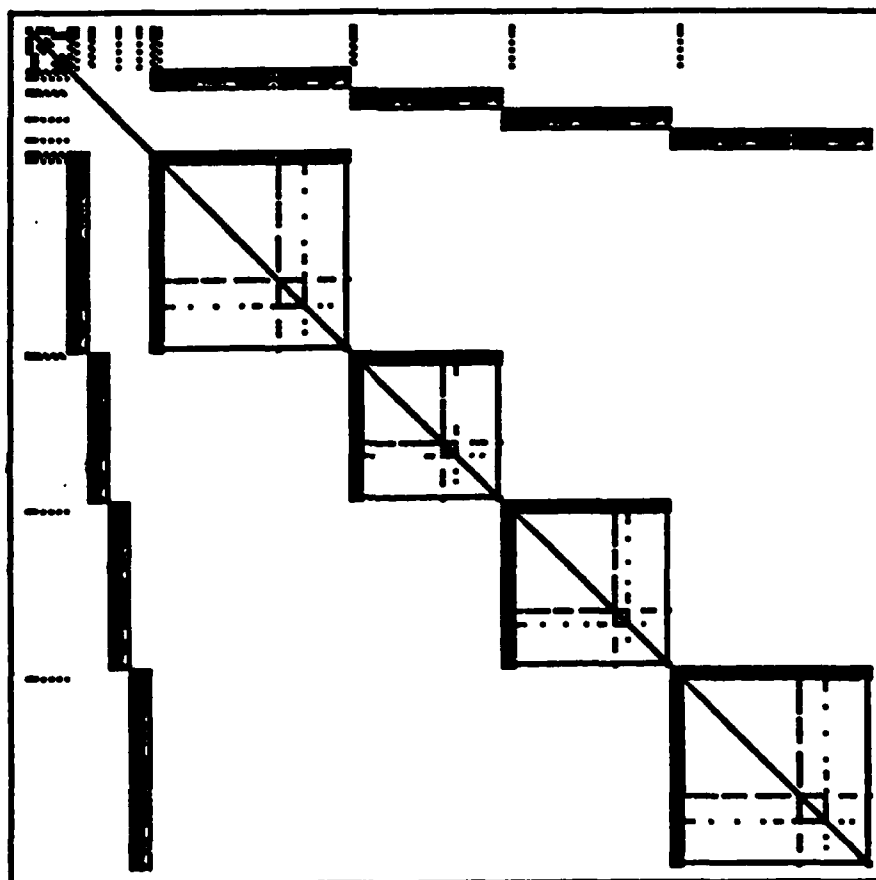
BANDM - Cholesky factors after YALE ordering



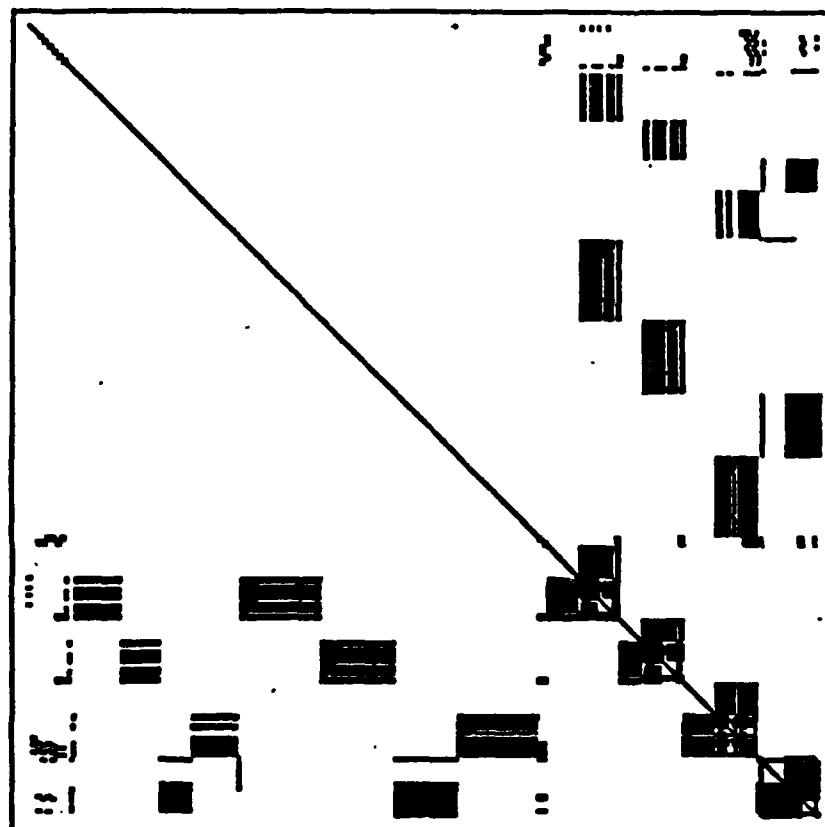
BANDM - Cholesky factors after Minimum Local Fill-in ordering



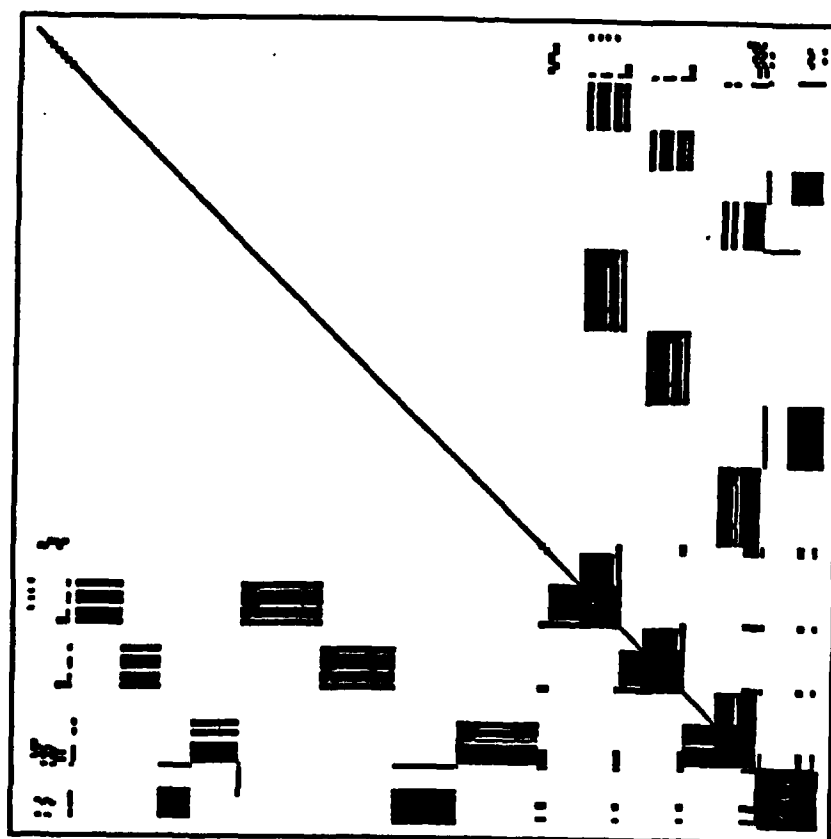
SHIP04L - LP matrix before cleaning



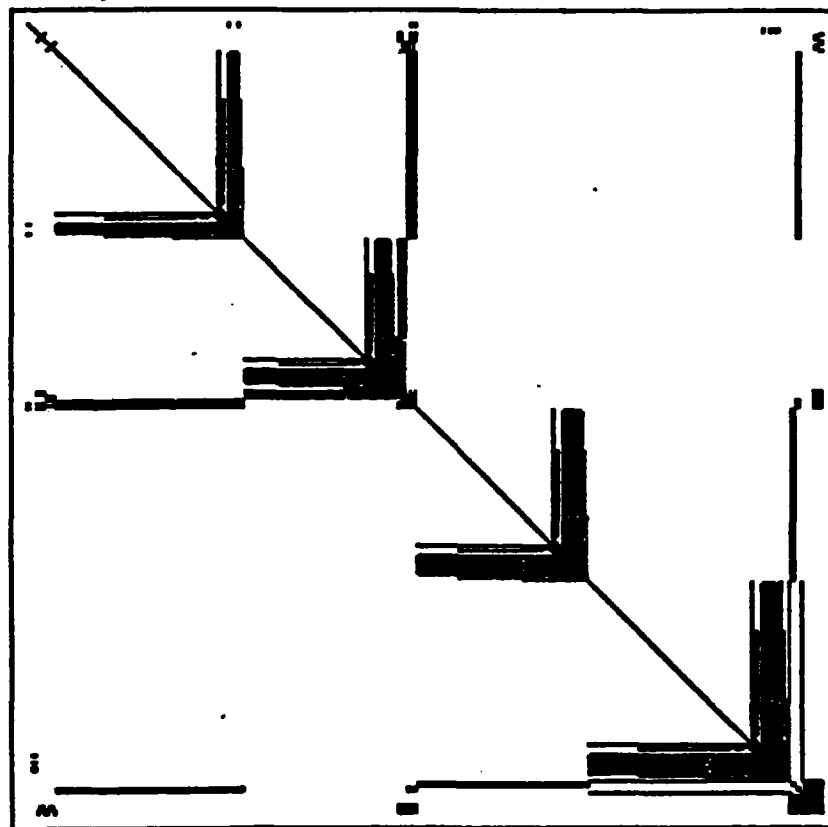
SHIP04L - AA^t before ordering



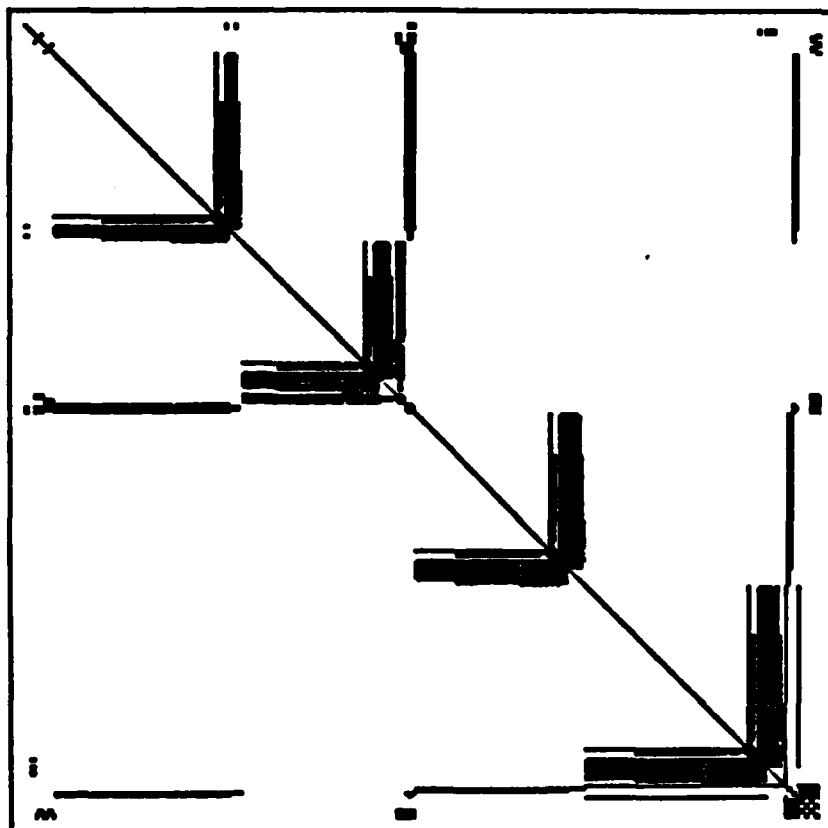
SHIP04L - AA^t after YALE ordering



SHIP04L - Cholesky factors after YALE ordering



SHIP04L - Cholesky factors after Minimum Local Fill-in ordering



SHIP04L - AA^t after Minimum Local Fill-in ordering

END

12-87

DTIC